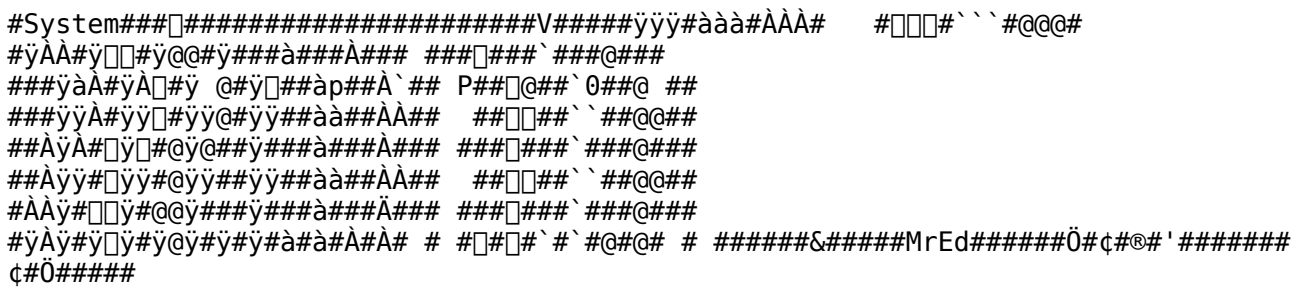

####ÿ□#ÿÿd#########d#2#######3##□###úðû

¤þÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#2#######½###This course provides an overview of the requirements activities (identification, analysis and management) necessary to produce software-intensive systems that meet the need of the end-user.t

##ÿ□#ÿÿd#########d#2#######3##□##ü÷##þ
}#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#2#######a###1. Identify the requirements activities that occur throughout the systems development life cycles

#a#########a#################a#########a####ÿ□#ÿÿd#########d#2#######3##□##ü÷{#□
Õ#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#2#######e###2. Identify characteristics and
types of good system/software requirements and ways to document theme

#e#########e#################e#########e####ÿ□#ÿÿd#########d#2#######3##□##ü÷w##
#Ñ#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#2#######D###3. Discuss the effects of requirements on project cost and schedule

çD¨#######D#¨#####3##□##bö□üþ F#ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######s###
Systems Requirements Analysis Software Requirements Analysis Requirements
Baselining Requirements ManagementD

#□F<#######PI<#####3##□##v÷-ür###ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######¾###IF the perfect document exists, it would be: Correct Unambiguous Complete Verifiable (Testable) Consistent Understandable by non-computer specialists Modifiable Traceable Annotated

#ÇG##a#T÷□ýÄ#$ÿ###########ýÿ##L
□F(#########(#####3##□##□÷ý□##ÿÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######[### Every requirement stated within it represents something required of the system to be builtd

#########aC¼øPøl ¤ú############ýÿ##ü#÷U##a# ö□ýØ
4############ýÿ##t#ÏB*#########*#####3##□##Úöý□
##ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2########### Every requirement has only one interpretationTerms with multiple meaning must appear in a glossary Potentially ambiguous words should be avoided or clarified Using formal notation reduces ambiguity but may be difficult to understand Diagramming may reveal ambiguity

```
#########aC¼øPøl

¤ú############ýÿ##ü#÷U##a#T÷□ýÄ#;#############ýÿ##ô##Yþ#########þ#####3##□##□÷-ý□###ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######Á###
```

Possesses the following four qualities Contains everything that the software is supposed to do Definitions of the responses of the software to all realizable classes of input data in all realizable classes of situations are included All pages are numbered; all figures and tables are numbered, named and referenced; all terms and units of measure are provided; all referenced materials are present No sections are marked □To Be Determined□ (TBD#

#########aC¼øPøl □ù############ýÿ##ü#÷U##a#T÷□ýÄ#~#############ýÿ###
ÏBp#########p#####3##□##□÷ý□#a#ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######Ã### Every requirement is verifiable; i.e., there exists some finite cost effective process with which a person or machine can check that the actual as-built software meets the requirement (testable).

#########aC¼øPøl ¤ú############ýÿ##| ÷F##a#T÷□ýÄ#<############ýÿ##d
çDH#########H#####3##□##□÷ý□###ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2###########
Characteristics for Understanding the requirements document are present: Logically organized Clear, precise wording No extraneous phrases Terms and abbreviations defined in glossary Use of graphics clarify rather than confuse Redundancy may exist to improve readability

```
#########aC¼øPøl  ¤ú############ýÿ##d#çD##a#T÷□ýÄ#(#############ýÿ##l
      #L2#########2#####3##□##□÷ý□###ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######□###
```

The origin of each requirement is clear It facilitates the referencing of each requirement in future development or enhancement document

#########aC¼øPøl

¤ú############ýÿ##\#Ç9##a#T÷□ýÄ#0#############ýÿ##<#÷Fb#########b#####3##□##□÷-
ý□###ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########d#2#######□### The document provides guidance
to the development organization Relative necessity (priority) Relative stability
Notes section

```
#########aC¼øPøl  □ù############ýÿ##D#O*##a#□÷@ü#     ?
#############ýÿ##\#Ç9□##########□#####3##□##Ê÷]üÆ#"#ÿÿ####ÿÿ####UÿÖ#ÿÿ□##########d
#2#######%### System capability (functional) requirements External interface
requirements Internal interface requirements System internal data requirements
Adaptation requirements Safety requirements Security and privacy requirements
System environment requirements Computer resource requirements
```

ÇG##a#À÷Üü0

Ø#############ýÿ##¤#÷F<#########<#####3##□##ú÷ùüö#»#ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########
d#########ì### System quality factors Design and construction constraints
Personnel-related requirements Training-related requirements Logistics-related
requirements Other requirements Packaging requirements Precedence/criticality
requirements

#########aC¼øPøl □ù############ýÿ##4
÷F##a#À÷Üü0

ð#############ýÿ##□#÷F@#########@#####3##□##ú÷ùüö#Ó#ÿÿ####ÿÿ####UÿÖ#ÿÿ□#########
d#########`###Why are better requirements needed?Better requirements lead to and result in better estimates

```
#########aC|ùPø,
□ù###########ýÿ##$#O*##a#Xöüýx
      é#############ýÿ##ü#ÇG2#########2#####3##□##□ö#þ>
      Ì#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######m### Consistency  Documentation
and History of Estimates Continuing effort to bring estimates closer to actuals#
```

#########aCdù□ø#
Óù############ýÿ##$#O*##a#àõ#ý#

ð#############ýÿ##t#□FT#########T#####3##□###ö#ýÆ#Ó#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########
Z#2#######s### Preliminary Cost Estimate (Quick Look) Proposal Estimate Software Development Plan Estimate Rebaseline Estimate#

```
#########aC□ù□ø8
Çù############ýÿ##<#ÇG##a#pöpü□
      l#############ýÿ##Ü#□Fö#########ö#####3##□##ªö□üV
      O#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######¦### Professional Judgment     □My
experience and training tell me...□ Analogy    □This job is similar to another
one...□ Automated Tools □The model results show that...□
```

```
#########aC ù□øP
ßù############ýÿ##$#O*##a#pöpü□
	f#############ýÿ##¤#□D□#########□#####3##□##ªö□üV
	I#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######w### System Experts Task breakdown
and effort estimate Tools (Swan) Schedule Driven  The Hat Method □And then Add
xx%□#
```

#########aC|ùÔø,
#ú############ýÿ##ü#÷U##a#pöpü################ýÿ##Ä##Y #########
#####3##□##ªö□üÖ
ó#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2########### Estimate using at least two techniques Validate models/productivity rates against knowledge, experience, common sense Get estimates from independent sources Adjust for the people doing the work Do it as a team, and document your assumptions and rationalen

```
#########aC(ù□øØ
      ßù############ýÿ##$#O*##a#pöpüà#f#############ýÿ##D#çD##B#3#dÿ□  □
############ýÿ####§Nx#########x#####3##□##ªö□ü¦#I#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z
#2#######`### Requirement Definition Requirements Control Process Models Mil-
Std-498  FP Analysis COCOMO
```

#########aCpùð÷
Dú###########ýÿ##$#O*##a#(ö0ýH
*#############ýÿ##$#O*H#########H#####3##□##böMý#
#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######,### B.W. Boehm, □Software Engineering Economics□, 1981 An accepted Industry Standard Based upon extensive research of previous information system projects Based upon software size (new and adapted) and 15 environmental cost drivers An Exponential Model - □Twice as big means more than twice as hard□

#V#########V#################V#########V####ÿ□#ÿÿd#########Z#2#######3##□##□÷□ÿÎ
V#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######F### The model allows comparison of the current project to an extensive calibration database of completed projects It is as if we had completed these projects ourselves and had metrics for them It is repeatable - thus, it fosters consistency It is recognized world-wide as an industry standard mechanism for software estimation

##ÿ□#ÿÿd#########d#2#######3##□##Âö#ý#
¿#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######M### BASIC: Considers only size of the software system INTERMEDIATE: Based upon software size (measured in Lines of Code [LOC]) plus 15 general □cost drivers□ that relate to the Product, Project, Personnel, and Environment complexity DETAILED: Intermediate COCOMO taken to the software component level. Complex and labor intensive.l

#M#########M#################M#########4###T Ö###□#########d#######µ###T Ö###□##
#######d#######d###T Ö###□#########d#########3##□##
÷=#& ª#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#2#######}###DSDC has chosen Intermediate COCOMO as a good compromise between accuracy and effort and time required to obtain estimationst

÷F##a#Põ□#°þ¸#############ýÿ##T#O*##¡#²þFû¦#fü############ýÿ##t

O*V#######duV#####3##□###öUþúþ»#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#########h###DEVELOPMENT PHASES Plans/Requirements Mgmt Product Design Programming Developmental Integration/Test

```
#h#########################V#################h#############T Ö###□#########d####
#######U Ö###□#########d###########U Ö###□#########d##########U Ö###□#########d#
##########U Ö###□#########d#########3##□##v#Aýb
      ##ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#########£### ACTIVITIES Requirements
Analysis Product Design Programming Developmental Test Planning
Verification/Validation Project Office CM/QA Manuals/Documentation#
```

##ÿ□#ÿÿd#########d#2#######3##□##òöiü#
Öþÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######□###All software estimation models exploit the correlation between software □size□ and the resultant effort/schedule required to produce it.

#□#########□#################□#########□####ÿ□#ÿÿd#########Z#2#######3##□##R÷Éÿn

□#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######<### □Size□ can be measured in just about any units, so long as one is consistent. (# listing pages, weight of listing, etc) The common size unit is Source Lines of Code (or equivalently, Delivered Source Instructions) Arguments rage over lines of code, but as long as one is consistent, they are pretty much academic

```
############################################ÿ□#ÿÿd#########d#2#######3##□###÷□ü2
      úþÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######§###Function Point Analysis (FPA)
attempts to solve this problem by modeling ultimate software size as a function
of the characteristics and attributes of the requirements#
```

#§#########§#################§#########§####ÿ□#ÿÿd#########Z#2#######3##□##j÷□#□
      ð#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######W### Counts of inputs, outputs, files, interfaces, and inquiries, ranked by complexity (simple, average, complex) Rankings on 14 □Influence Factors□ which describe the overall software product A pre-calibrated mapping of function points to Lines of Code (LOC) for a variety of programming languages in common usage (and some not-so-common ones!)#

##ÿ□#ÿÿd#########d#########1##□##2úCüT
#Oýÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#########p###An objective, quantitative measurement of the size andcomplexity of a system based upon the user point of view.

##ÿ□#ÿÿd#########d#2#######1##□##lü□ýN #ß#ÿÿ####ÿÿ####Aÿ□#ÿÿd#########d#########□### Required Software Reliability Data Base Size Product Complexity Execution Time Constraint Main Storage Constraint Virtual Machine Volatility Computer Turnaround Time Analyst Capability Applications Experience Programmer Capability Virtual Machine Experience Programming Language Experience Modern Programming Practices Use of Software Tools Required Development Schedule

```
############################################ÿ□#ÿÿd#########d#2#######3##□##\öiüb
      F#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#########Ð###Two well established software
engineering principles:If you compress a software development schedule, the work
will require more effort (staff hours)If you have □all the time in the world,□
you will take it
```

#Ð#########6###############□#################Ð#########6####ÿ□#ÿÿd#########d#2##
###a####ÿ□#ÿÿd#########Z#######9####ÿ□#ÿÿd#########Z#########3##□##ªöé#¶
ö#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######Þ### Compression of the schedule leads to higher risks Compression to less than about 70% of nominal is generally unrealistic. One effective mechanism for schedule compression is to compress only the critical delivery tasks.

#Þ#########½###Þ#########3###U Ö##
#□#########d#######I###U Ö###□#########d#######b###U Ö###□#########d#########3##
□##Úö±#&   O#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########d#2#######l###COCOMO does not lower the intrinsic risk of schedule compression, BUT it does make it quantitatively visible

##ÿ□#ÿÿd#########d#2#######3##□##ªö□ýö #K#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######Í### An estimate of Effort (staff-months) for the project An estimate of the nominal Schedule (months) A distribution by development phase and activity An estimated average staff size, based on productivity#

#Í#########Í#################Í#########6###U Ö###□#########d#######.###U Ö###□##
#######d#######2###U Ö###□#########d#######7###U Ö###□#########d#########3##□##z
öM#ö#□#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#########×###COCOMO assumes □commercial
documentation,□ and certain other characteristics about the software engineering
methodologies. The raw results of the model can be scaled to accommodate
variations from these assumptions#

####ÿ#ÿÿd#########d#2#######3####öíüz
Zÿÿÿ####ÿÿ#####ÿ#ÿÿd#########d#2#######u###DSDC provided data for a number of completed projects. These data were used to create estimates from various models.t

#u#########u#################u#########u####ÿ□#ÿÿd#########Z#2#######3##□##òö□#>
C#ÿÿ####ÿÿ#####ÿ□#ÿÿd#########Z#2#######,### Intermediate COCOMO estimates effort for Organic Mode projects with very low errors (average < 1%) Schedule estimates tends to be a little inflated, which is highly typical Large project data provided an extremely good fit, however number of projects was insufficient for statistical significance

```
# ######### ################# ######### ####ÿ#ÿÿd#########d#########2####ú÷ü¦
s#ÿÿ####ÿÿ#####ÿ#ÿÿd#########d#########R###Click to edit Master text styles
Second LevelThird LevelFourth LevelFifth Level
```